

# NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



## THESIS

### IMPROVED COMPUTER MODELING OF SHIP PROGRESSIVE FLOODING AS A DESIGN TOOL

by

Russell A. Wright II

September 1998

Thesis Advisor:

Fotis A. Papoulias

Approved for public release; distribution is unlimited

19981215 129

DTIC QUALITY INSPECTED 4

| REPORT DOCUMENTATION PAGE  |  |  | Form Approved OMB No. 0704-0188                |   |
|--|--|--|--|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to: Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. |  |  |  |   |
| 1. AGENCY USE ONLY (Leave blank)   |  | 2. REPORT DATE<br>September 1998                         |  | 3. REPORT TYPE AND DATES COVERED<br>Master's Thesis |
| 4. TITLE AND SUBTITLE Improved Computer Modeling of Ship Progressive Flooding as a Design Tool   |  |  | 5. FUNDING NUMBERS                             |   |
| 6. AUTHOR(S) Russell A. Wright II  |  |  |  |   |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>Naval Postgraduate School<br>Monterey, CA 93943-5000   |  |  | 8. PERFORMING ORGANIZATION REPORT NUMBER       |   |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  |  |  | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |   |
| 11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.   |  |  |  |   |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT<br>Approved for public release; distribution is unlimited   |  |  | 12b. DISTRIBUTION CODE                         |   |
| 13. ABSTRACT (maximum 200 words)<br><p>When a ship suffers underwater damage, there is a rapid influx of water, followed by a period of slower progressive flooding. This results in flooding of compartments whose hull boundaries, but not interior bulkheads, are still intact. An existing computer model uses the FORTRAN computer language and formatted input files to model progressive flooding. This thesis uses MATLAB computer language and SIMULINK graphical user interface to provide a modular, expandable progressive flooding design tool.</p>   |  |  |  |   |
| 14. SUBJECT TERMS PROGRESSIVE FLOODING, COMPUTER MODEL   |  |  | 15. NUMBER OF PAGES 58                         |   |
|  |  |  | 16. PRICE CODE                                 |   |
| 17. SECURITY CLASSIFICATION OF REPORT<br>Unclassified, Official use Only   | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>Unclassified, | 20. LIMITATION OF ABSTRACT<br>UL               |   |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. Z39-18 298-102



Approved for public release; distribution is unlimited.

**IMPROVED COMPUTER MODELING OF SHIP PROGRESSIVE  
FLOODING AS A DESIGN TOOL**

Russell A. Wright II  
Lieutenant, United States Navy  
B.S., University of Florida, 1990

Submitted in partial fulfillment  
of the requirements for the degree of


**MASTER OF SCIENCE  
IN  
MECHANICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**

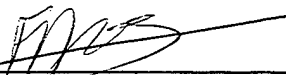
September 1998

Author:



Russell A. Wright II

Approved by:



Fotis A. Papoulas, Thesis Advisor



Terry McNelly, Chairman  
Department of Electrical and Computer Engineering



## **ABSTRACT**

When a ship suffers underwater damage, there is a rapid influx of water, followed by a period of slower progressive flooding. This results in flooding of compartments whose hull boundaries, but not interior bulkheads, are still intact. An existing computer model uses the FORTRAN computer language and formatted input files to model progressive flooding. This thesis uses MATLAB computer language and SIMULINK graphical user interface to provide a modular, expandable progressive flooding design tool.



## TABLE OF CONTENTS

|  |    |
|--|----|
| I. INTRODUCTION.....                                       | 1  |
| A. MOTIVATION.....   | 1  |
| B. BACKGROUND.....   | 2  |
| C. DEFINING THE MODEL.....                                 | 2  |
| D. SIMULINK.....   | 4  |
| E. OBJECTIVES.....   | 4  |
| II. SIMULINK PROGRAM.....                                  | 7  |
| III. PROGRAMS AND SUBROUTINES.....                         | 9  |
| A. FLOW RATE THROUGH HOLE(S).....                          | 9  |
| B. NET FLOW INTO A COMPARTMENT.....                        | 11 |
| C. HEIGHT OF WATER (HOW).....                              | 11 |
| D. MOMENTS DUE TO ADDED WATER.....                         | 11 |
| E. KG (VERTICAL POSITION OF CENTER OF GRAVITY).....        | 13 |
| F. VERTICAL CENTER BOUOYANCY (KB).....                     | 13 |
| G. HEIGHT OF METACENTER ABOVE CENTER OF BUOYANCY (BM)..... | 16 |
| H. RIGHTING ARM (GM).....                                  | 18 |
| I. HEEL AND TRIM ANGLE.....                                | 18 |
| J. ADDITIONAL SUBMERGENCE.....                             | 23 |
| K. PROGRAM USAGE.....                                      | 23 |
| IV. RESULTS.....   | 25 |
| V. DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS.....        | 29 |
| A. DISCUSSION.....   | 29 |

|                                |    |
|--------------------------------|----|
| B. CONCLUSIONS.....            | 29 |
| C. RECOMMENDATIONS.....        | 29 |
| APPENDIX A HULL.DAT.....       | 31 |
| APPENDIX B. ARVOL.M.....       | 43 |
| APPENDIX C. DRAFTS.M.....      | 45 |
| LIST OF REFERENCES.....        | 47 |
| INITIAL DISTRIBUTION LIST..... | 49 |

## I. INTRODUCTION

### A. MOTIVATION

Accurate modeling of progressive flooding is essential in assessing the survivability features of a given ship. A realistic flooding and counter flooding scenario is a very complex dynamic system including complicated vessel motions, time-varying external excitation, changing hull mass and hydrodynamic properties, progressive flooding through damage openings, internal sloshing of flooded water, and de-watering control actions. Although, due to the complexity of the problem, the traditional method of approach is through large amplitude static analyses, the need for a more accurate analysis has been recognized. In order to bridge this gap, recently there has been a large amount of research focusing on dynamic stability prediction taking into consideration water-on-deck and large amplitude time-dependent motions and excitation [Ref. 1]. Equally important, however, is the development of a design methodology for incorporating elements of man/machine interface into the problem; in other words effects of decision-making and control strategies. This presents an additional level of difficulty since combinations of discrete events and continuous differential equations describe such systems. The long term goal that is initiated with this thesis is to study the development of an improved damage stability analysis and design tool, which will address the effects of control decisions and vessel dynamics in a progressive flooding scenario.

## **B. BACKGROUND**

When a ship suffers underwater damage, there is a rapid influx of flooding water, usually followed by a period of slower progressive flooding. Progressive flooding is defined as flooding which extends beyond the compartments that are open directly to the sea. This results in flooding of compartments whose hull boundaries, but not interior bulkheads, are still intact. Existing computer models at the Naval Postgraduate School utilize the FORTRAN computer language and formatted input files to model progressive flooding [Ref. 2]. Any change in model design conditions, flooding damage, or damage control abilities requires a completely new FORTRAN input file. The purpose of this thesis is to include the capabilities of the original computer progressive flooding model in a more modular and expandable format, using the user-friendly MATLAB and SIMULINK programs.

## **C. DEFINING THE MODEL**

### **1. Hole Model**

Holes can be modeled as a short-tube orifice as shown in figure 1 with a diameter (hole size) much larger than its tube length (ship's skin thickness). The flow discharge coefficient for this case is 0.816 [Ref.3: pp.42-43].

### **2. Hull Model**

The required hull characteristics are determined using the same table of offsets used by Dawson (Hull.dat) [Ref. 2: pp. 127-143] and MATLAB programs `arvol.m` and `drafts.m` to calculate the locations of center of buoyancy and center of floatation, waterplane area, and waterplane moments used in the program equations. Both Matlab

programs are included in the appendix. For the FFG-7 hull model used here the following values were calculated:

Table 1: Hull Model Characteristics

Center of Buoyancy

|           |                        |
|-----------|------------------------|
| Xbar..... | 209.9 ft aft of FP     |
| Ybar..... | 0.0 ft from CL         |
| Zbar..... | 10.52 ft from baseline |

Center of Floatation

|           |                    |
|-----------|--------------------|
| Xbar..... | 227.4 ft aft of FP |
| Ybar..... | 0.0 ft from CL     |

Waterplane area.....14030 ft<sup>2</sup>

Longitudinal Moment of Inertia.....1.5883 × 10<sup>9</sup> ft<sup>4</sup>

Transverse Moment of Inertia.....1.8148 × 10<sup>6</sup> ft<sup>4</sup>

### 3. Equilibrium Model

The added weight method [Ref.4: pp.76-79] is used within the program to determine the equilibrium condition (displacement, heel, and trim) of the hull model for any state of flooding. This method assumes hull characteristics are constant and is accurate for “small angles”. The added weight method treats flooding water as added weight and treats the hull as if it were intact. The effect of the added weight is divided into a weight, which causes the ship to immerse evenly (parallel sinkage) and moments which cause the ship to list and trim about its center of floatation. The small angle assumption assumes that the ship lists and trims about a fixed center of floatation. In reality, since the hull is not cylindrical, the location of the center of floatation will change

with changing list and trim. This method allows much simplified calculation of hull characteristics compared with the lost buoyancy method. Since the weight added is liquid water which is free to move as the ship moves (and thus does not contribute to pitch and roll resistance), a free surface correction is added to the longitudinal and transverse moments of inertia to account for the reduced resistance of the hull to pitch and roll.

#### **4. Compartment Model**

The ship compartments are modeled as rectangular boxes. Extension to more realistic shapes is one of the recommendations for further expansions to the capabilities of the program.

#### **D. SIMULINK**

The SIMULINK program provides a graphical representation of mathematical relationships and equations to enable the user to determine the response of system models. Graphical symbols and connections are used instead of numbers and mathematical symbols, allowing more intuitive use and understanding of complex mathematical relationships between system components. The SIMULINK program also allows easy modification of input parameters and import and export of data from MATLAB programs. Lastly, SIMULINK allows the user to observe system or individual component behavior during the simulation through the use of output graphs.

#### **E. OBJECTIVES**

The objective of this thesis is to develop a progressive flooding model to utilize the superior user interface and data exchange capabilities of the SIMULINK program. The

program must be able to accurately calculate flooding rates and the subsequent response of the ship. The program must also be user-friendly, easily modified and have the potential for further expansion by subsequent users. To achieve these goals the following programs were written by the author:

## **1. Programs**

### **a. Pflood.m**

Pflood.m is the main program created by the author. This program calculates the flooding and ship response caused by a hole of user-chosen size and location in the ship's hull. The flooding rate, flooded volume, heel and trim are calculated and updated each second. In addition, the user can specify the number, time of activation and size of pumps used for dewatering and the duration of their use.

### **b. Arvol.m**

Arvol.m is a MATLAB program written by the author to determine the center of buoyancy of the hull model defined by the Hull.dat data file using Simpson's rule integration techniques.

### **c. Drafts.m**

Drafts.m is a MATLAB program written by the author to determine the waterplane area, center of floatation, and second moment of inertia of the hull waterplane for a given draft.

## **2. Validation**

Program performance was validated by comparing the program output to expected ship behavior for a given flooding condition.



## II. SIMULINK PROGRAM

The schematic representation of Pflood.m is shown in Figure 1. For illustrative purposes, Pflood.m is shown in its "expanded" form. By leaving Pflood.m in expanded form, any function or variable can be observed or changed by the user. For ordinary use, the various subroutines would be grouped into subsystem blocks with only the user-defined variables and the outputs represented by discrete function blocks. Placing the program in subsystem block form has the added advantage of greatly enhancing program readability and user-friendliness. Specific details of the individual block diagram elements are included in Chapter III.



### III. PROGRAM AND SUBROUTINES

#### A. FLOW RATE THROUGH HOLE(S)

The flow through the damage, either primary or secondary, can be modeled as turbulent flow through an orifice of negligible length, or a hole. Flow rate through a hole is a function of the hole size (cross sectional area), hole shape, and pressure difference across the hole. The hole size and shape are related to the discharge coefficient,  $C_d$ . A  $C_d$  of 0.816 is used in this application in order to remain consistent with Dawson's previous computer flooding model. This number is also consistent with theoretical predictions of flow through orifices as explained in the previous chapter [Ref. 3: pp. 42-43]. The pressure difference is determined by the difference in height of the liquid on either side of the hole. The equation is

$$Q = C_d \times A \times \sqrt{2g \times (h_2 - h_1)}$$

where  $Q$  is the volumetric flow rate,  $A$  is the cross sectional area,  $C_d$  is the discharge coefficient,  $g$  is the acceleration due to gravity, and  $h_2 - h_1$  is the liquid height difference across the hole. This height difference is a function of the level of the flooding water inside the flooded tanks, as well as the draft and trim of the ship. The latter quantities will have to be calculated using the added weight method as explained in the previous chapter. The Simulink representation of the above equation is shown in Figure 2. The program allows continuous adjustment of the values of  $h_1$  and  $h_2$  to account for increased immersion of the exterior hole and increased compartment water height.

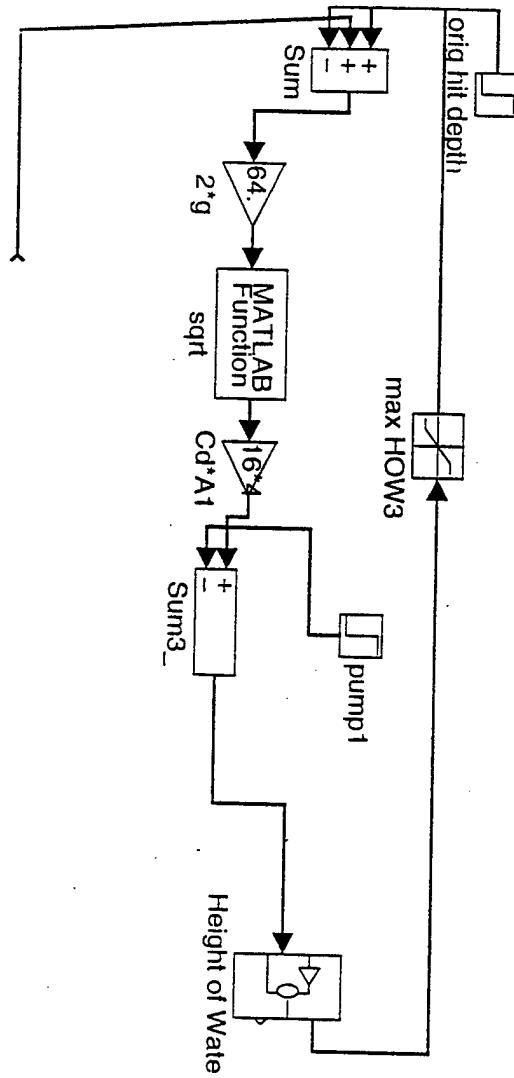


Figure 2: SIMULINK Hole Flow Symbolgy

## **B. NET FLOW INTO A COMPARTMENT**

Flooding water removed from the compartment by pump activation or by flow into adjacent compartments is subtracted from the water flow rate into the compartment. Two way flow due to water removal from one of two connected flooded compartments is also possible, and is accounted for by taking the square root of absolute value of the liquid height difference and then multiplying the result by the sign of the height difference. This method avoids program failure due to trying to take the square root of a negative number. This program also continuously updates the values of  $h_1$  and  $h_2$  to account for increased immersion of the exterior hole and increased compartment water height.

## **C. HEIGHT OF WATER (HOW)**

Once the net volumetric flow rate of water is known, integrating the flow rate results in the net volume of water added to the compartment. This volume, divided by the horizontal area of the compartment, and divided again by the permeability of the compartment, gives the height of water in the compartment. A saturation function ensures that the height of water in the compartment is always between zero and the maximum height of the compartment. Figure 3 shows the Simulink height of water function.

## **D. MOMENTS DUE TO ADDED WATER**

The added weight method treats the added weight as a combination of a weight which causes the ship to sink evenly (parallel sinkage) and transverse and longitudinal

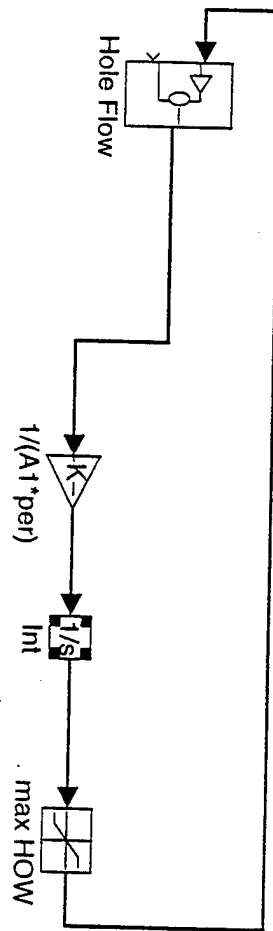


Figure 3: SIMULINK Height of Water Terminology

moments which cause the ship to heel and trim respectively. Multiplying the height of water (HOW) by the product of the weight density of the liquid ( $\rho g$ ), the compartment permeability (per), and the compartment horizontal area (A) determines the weight added to the compartment. Multiplying the added weight by the moment arm ( $Z_{trans/long}$ ) from the ship's center of floatation to the center of gravity of the added weight gives the moment about the center of floatation in the longitudinal and transverse directions. In equation form:

$$Moment = (\rho g A \times HOW \times per) \times Z_{trans / long}$$

The Simulink representation of the above equation is shown in Figure 4.

#### **E. KG (VERTICAL POSITION OF CENTER OF GRAVITY)**

KG (Vertical Position of Center of Gravity) is determined by multiplying the original weight of the ship by the original KG, adding the product of the weight of the added water and the height of the added water center of gravity above the keel, and dividing the sum by the new ship weight. The equation for KG is:

$$KG_{NEW} = (KG_{OLD} \times W_{OLD} + KG_{ADD} \times W_{ADD}) \div W_{NEW}$$

The Simulink representation is shown in Figure 5.

#### **F. VERTICAL CENTER OF BUOYANCY (KB)**

The ship's new vertical center of buoyancy (KB) is determined by multiplying the original KB by the original displacement, adding the product of the added weight and the original draft plus one half of the parallel sinkage, then dividing the product by the new

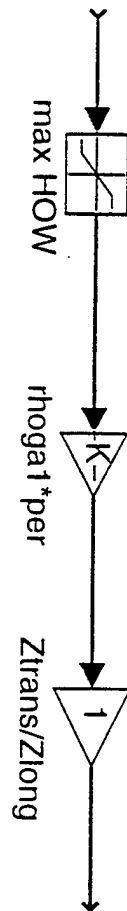


Figure 4: SIMULINK Moment Terminology

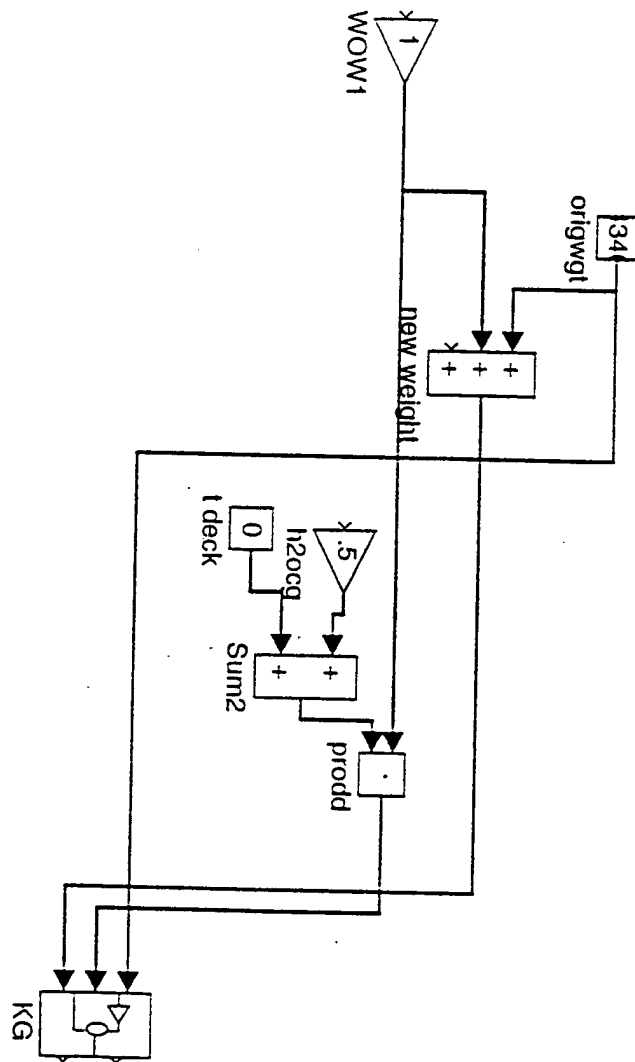


Figure 5: SIMULINK KG Terminology

weight. In equation form:

$$KB_{NEW} = (KB_{ORIG} \times W_{ORIG} + KB_{ADD} \times W_{ADD}) \div W_{NEW}$$

This is based on the assumption that the hull is effectively wall-sided, and that the vertical center of buoyancy of the newly submerged portion is located at one half the added draft.

Both of these assumptions are consistent with the small angle assumption that was mentioned in the previous chapter. If the small angle assumption is no longer true, then the true position of the center of buoyancy will have to be calculated. This can be done by employing numerical integrations and utilizing the provided table of offsets, or by interfacing the program with a standard external ship hydrostatics calculations program, such as SHCP [Ref. 2: pp. 24-25]. Such an interface is relatively straight forward since the computational environment of Matlab/Simulink already allows incorporation of external ASCII files. Figure 6 shows the Simulink representation of the KB equation.

#### **G. HEIGHT OF THE METACENTER ABOVE THE CENTER OF BUOYANCY (BM)**

The height of the metacenter above the center of buoyancy (BM) is determined by the second moment of the waterplane (about the waterplane centerline or longitudinal center of floatation) divided by the hull displacement. For the added weight method, a free surface correction is made by subtracting the second moment of the flooded compartment area from the waterplane moment. Typically, free surface corrections are done for liquids when calculating the final metacentric height. In such a case, the free surface correction appears as a reduction in the metacentric height or, equivalently, an increase in the virtual center of gravity. The same effect can be observed by modifying

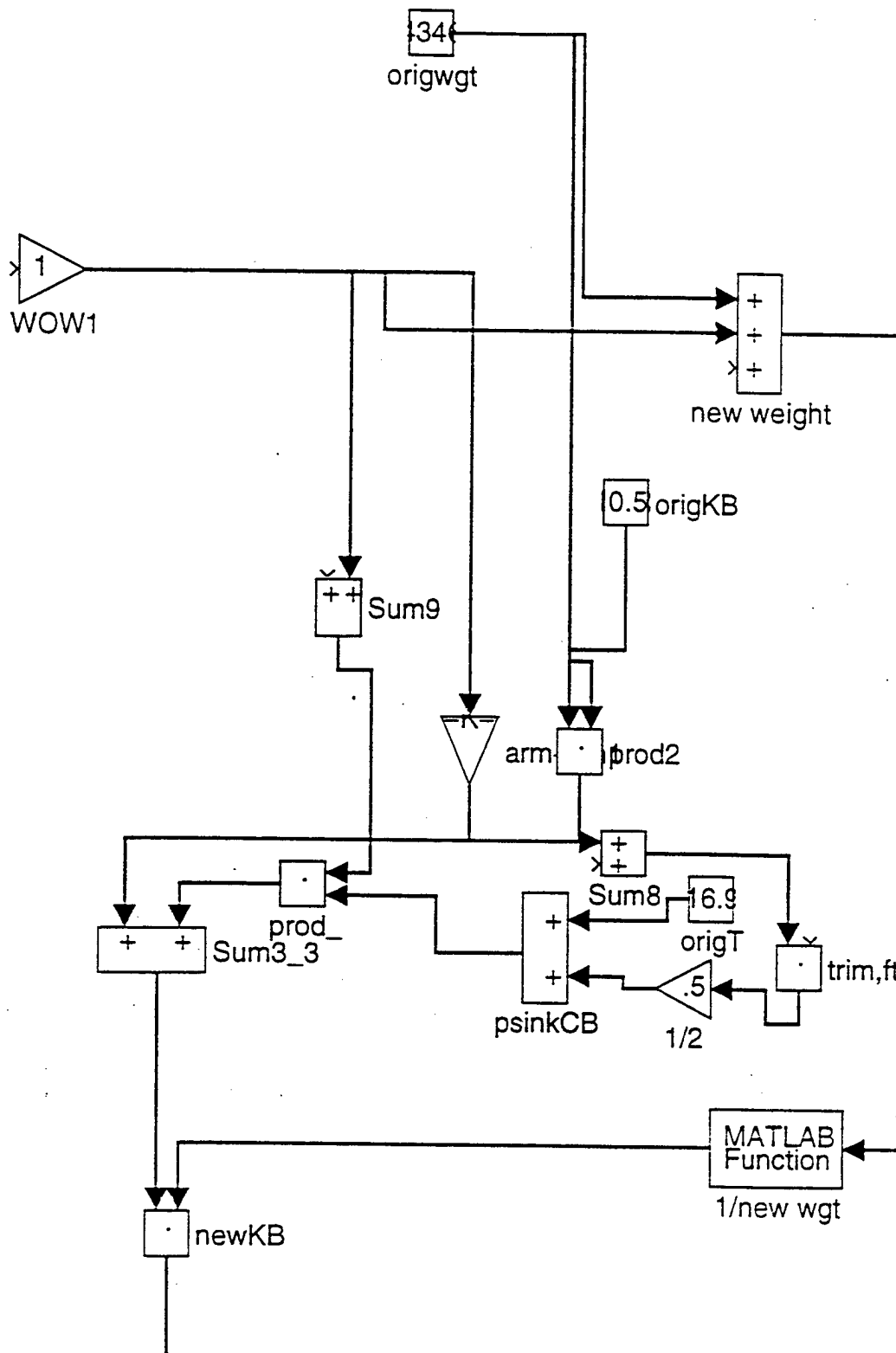


Figure 6 : SIMULINK KB Terminology

the second moment of the waterplane area in the calculation of the metacentric radius, BM. This is because the added weight in this case is seawater. With this modification in BM, the added weight method produces identical results with the lost buoyancy calculations (reference a naval architecture book). The BM Simulink terminology is shown in Figure 7.

## H. RIGHTING ARM (GM)

The righting arm (GM) is determined by  $GM = KB + BM - KG$ . This equation is graphically represented by the triple summing point shown in Figure 8.

## I. HEEL AND TRIM ANGLE

The angle of heel is determined by

$$\sin \phi = Volume \times Z_{TRANS} \div (Displacement \times GM)$$

as shown in Figure 9. The total trim is determined by dividing the total trimming moment by the moment to change trim (MCT). MCT is equal to the total weight times the longitudinal GM, divided by the ship length at the waterline. The tangent of the trim angle is equal to the total trim divided by the ship's length at the waterline. If the large angle assumption needs to be utilized, the moment to change trim will have to be imported as an external array. This is not expected to cause significant difficulties since MCT is readily available from the hydrostatic diagrams of a given ship. The Simulink trim angle terminology is shown in Figure 10.

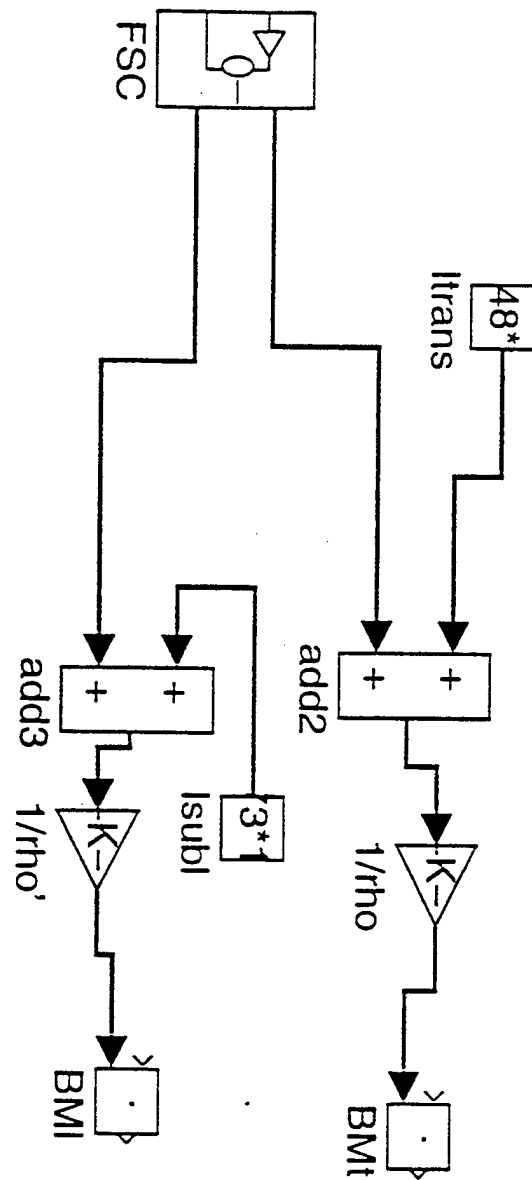


Figure 7: SIMULINK BM Terminology

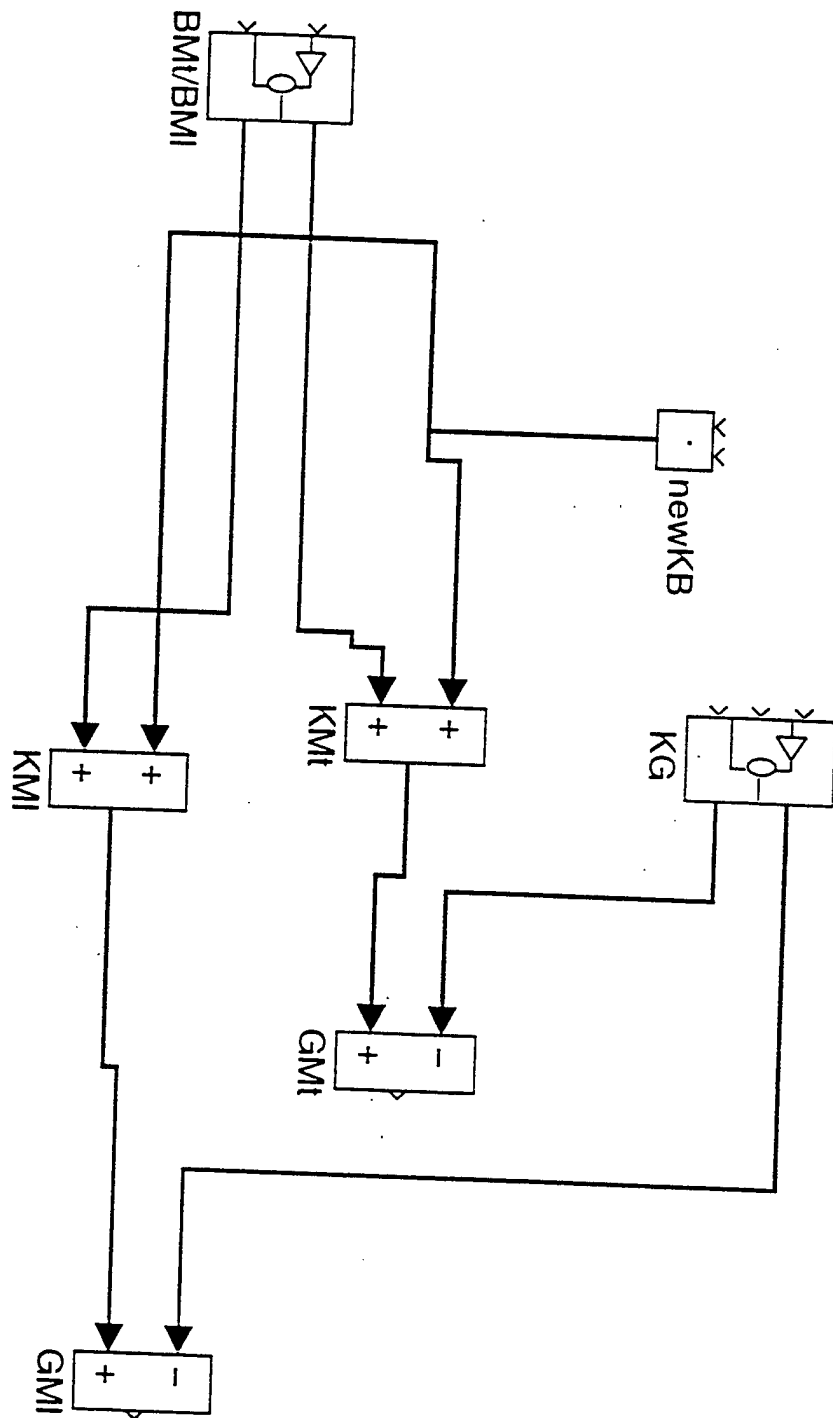


Figure 8: SIMULINK GM Terminology

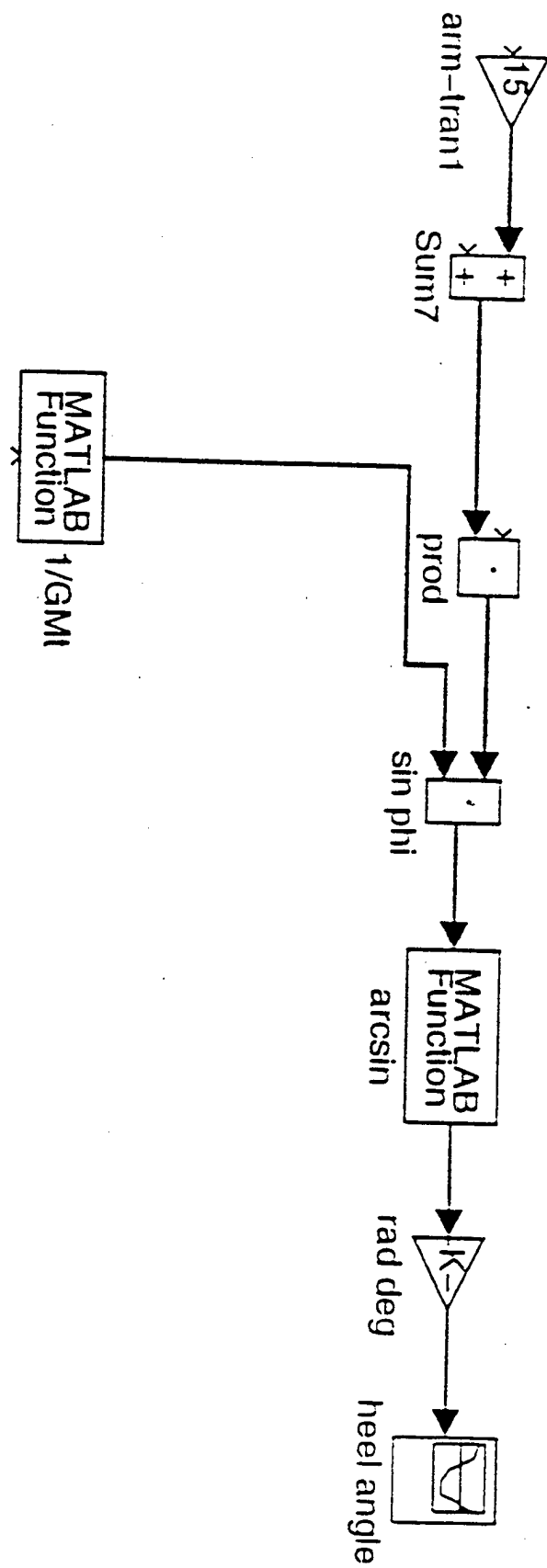


Figure 9: SIMULINK Heel Angle Terminology

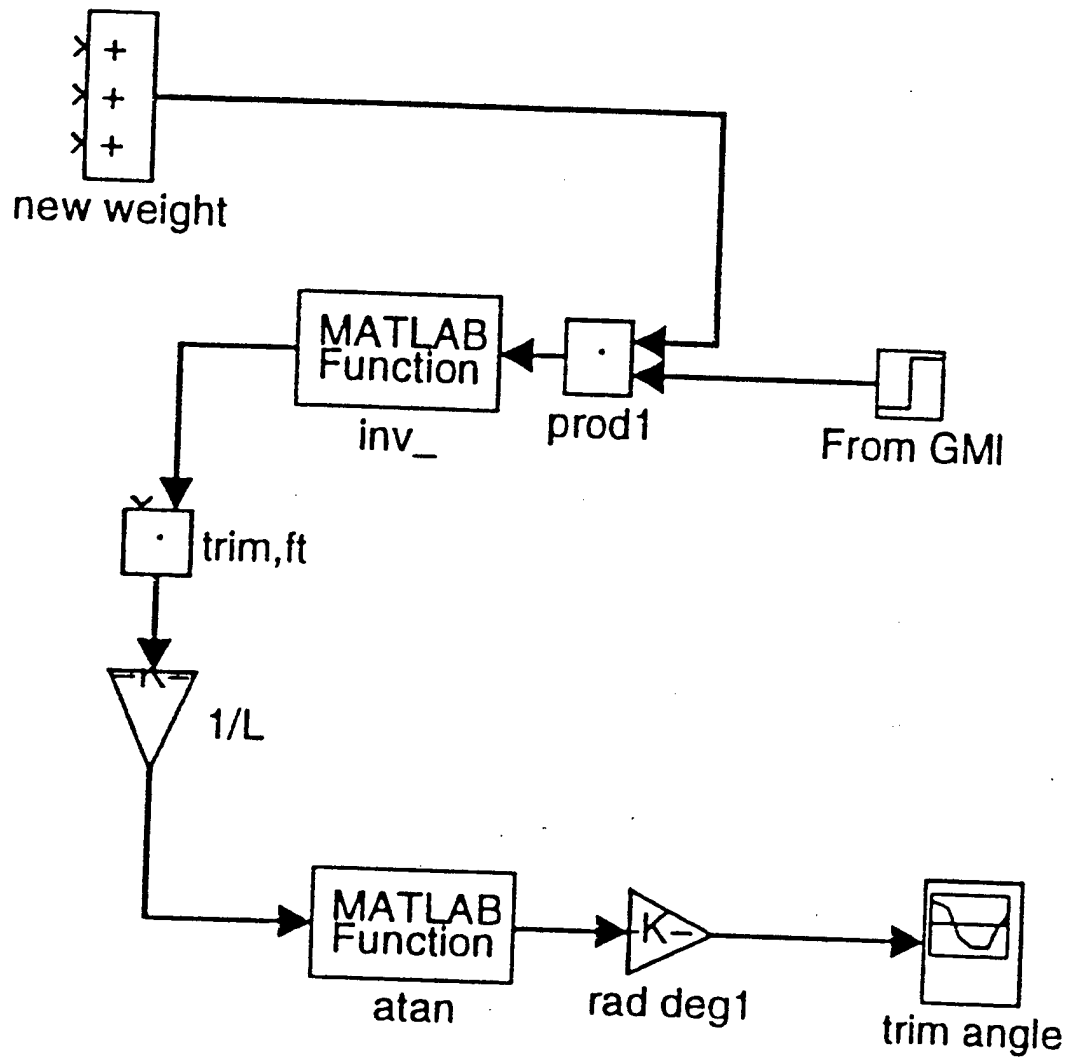


Figure 10: SIMULINK Trim Angle Terminology

## J. ADDITIONAL SUBMERGENCE

The additional submergence of the hole due to list is determined by multiplying the sine of the heel angle by the transverse distance from the transverse center of floatation to the hole. Similarly, the increased hole depth due to trim is obtained by multiplying the sine of the trim angle by the longitudinal distance to the center of floatation. The new hole depth is the sum of the original hole depth and the depth increases due to heel and trim. In algebraic form, these relations are shown by

$$h_{2NEW} = h_{2OLD} + (Z_{TRANS, hole} \times \sin \phi) + (Z_{LONG, hole} \times \sin(trim))$$

and in graphical form they are represented by the diagram of Figure 11.

## K. PROGRAM USAGE

To use Pflood.m, the user needs only select "simulation" from the SIMULINK menu, then select "start". Modifications to the simulation parameters or the program are easily accomplished by "double-clicking" on the symbol of interest. Additionally, SIMULINK allows the user to easily add output displays in order to observe the behavior of any desired variable during the simulation.

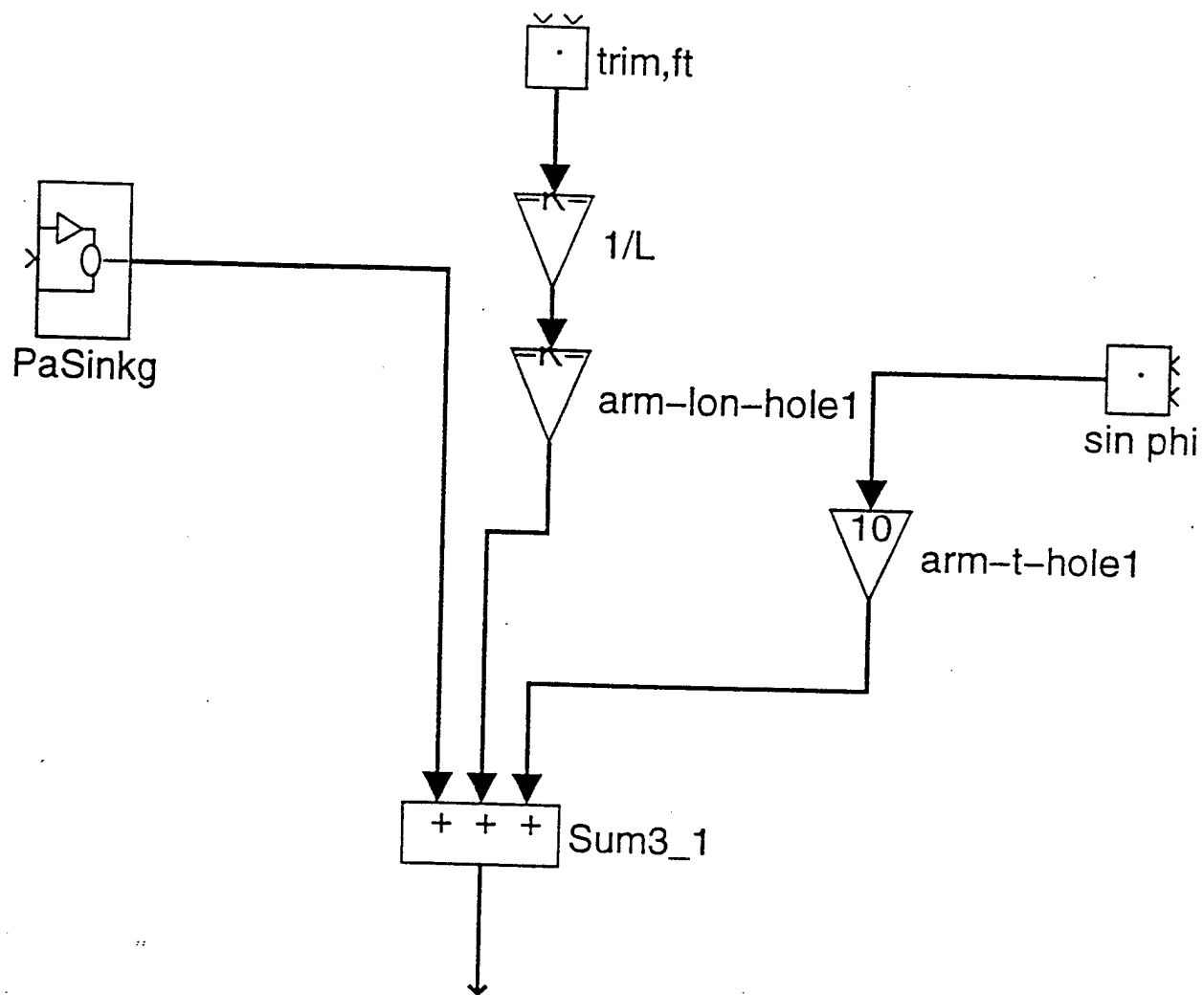


Figure 11: SIMULINK Hole Submergence Terminology

#### IV. RESULTS

Results for a test case consisting of a compartment with a 4 square foot hull penetration (compartment 1) and a 4 square inch watertight bulkhead penetration (compartment 2). Pumps were activated at 10 minutes (600 seconds) which completely dewatered compartment 1 in order to test the ability of the program to account for backflow through holes. As shown in figure 12, the four square foot hull penetration completely floods compartment 1 essentially instantly. Activating the pumps dewater compartment 1, instantly removing the water entering through the hull penetration. Similar results could have been achieved by changing the characteristics of the hull penetration to simulate hull repairs.

Figure 13 illustrates how the program accurately predicts backflow into compartment 1 from compartment 2, which had no pumps assigned. A saturation function ensures that backflow from compartment 2 will stop when compartment 2 water height drops to the level of the bulkhead penetration.

The predicted trim response of the ship is shown in Figure 14. The ship initially trims as a result of the rapid flooding of compartment 1, then gradually increases its trim angle as compartment 2 floods. The instant dewatering of compartment 1 then causes the trim angle to decrease, followed by further trim angle reduction as compartment 2 is dewatered. Without altering the pump or hole configuration, the ship will assume a steady trim angle corresponding to compartment 2 flooded up to the height of the bulkhead penetration.

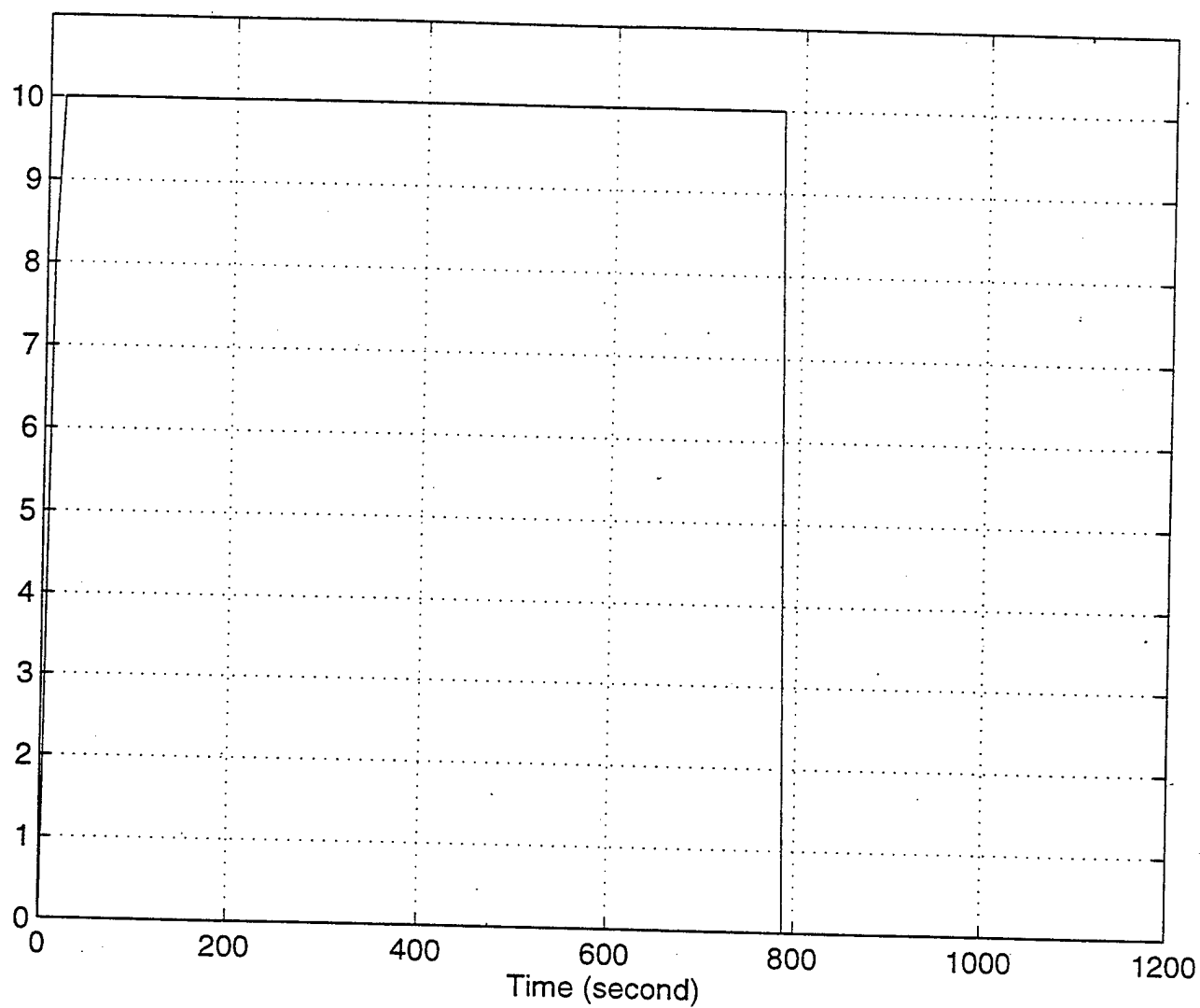


Figure 12: Height of Water in Compartment 1

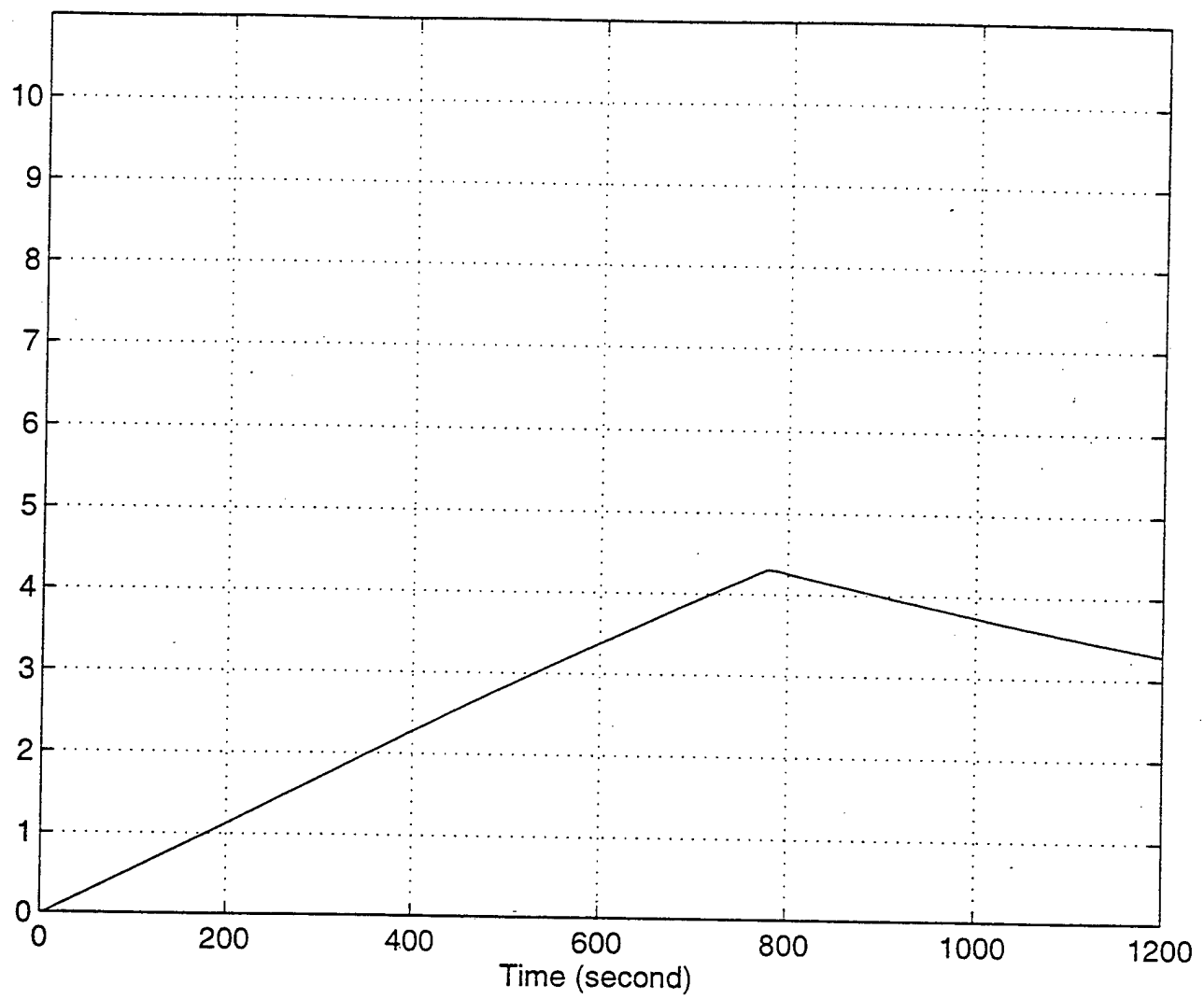


Figure 13: Height of Water in Compartment 2

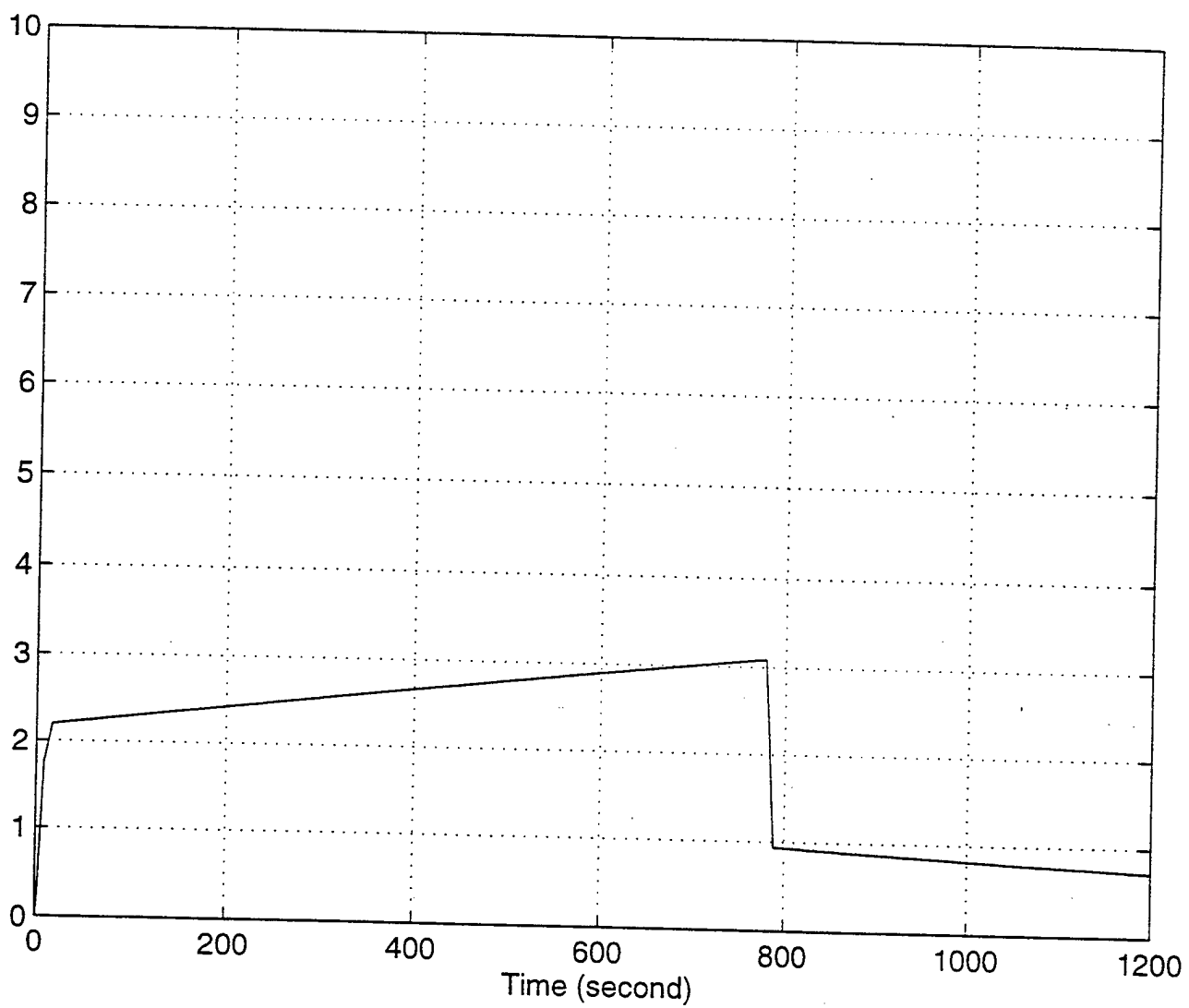


Figure 14: Ship Trim Response

## **V. DISCUSSION, CONCLUSIONS, AND RECOMMENDATIONS**

### **A. DISCUSSION**

This thesis has successfully developed the foundation for a modular and expandable progressive flooding computer model. The computer model accurately predicts the quasi-static ship response to progressive flooding and is easily modified by the user to explore ship response to any desired damage situations.

### **B. CONCLUSIONS**

Based on the above results, the following conclusions are drawn:

1. Pflood.m is an effective design tool for modeling and studying the effects of progressive flooding.
2. Pflood.m is easily modified to model virtually any damage scenario.
3. Pflood.m can be expanded to include different ship progressive flooding and equilibrium models.

### **C. RECOMMENDATIONS**

The following items are recommended to further expand Pflood.m and enhance its utility as a design tool:

1. Incorporate a hull characteristics look-up table into the Pflood.m program to expand the ship response modeling ability beyond "small angles" of heel and trim and non-rectangular compartments.
2. Explore the effect of pump configuration on dewatering effectiveness and ship response.



# APPENDIX A: HULL.DAT

```

-.98 .203 34.011
-.98 1.0097 37.035
-.98 2.09 40.059
-.98 2.30593 40.636
-.98 2.525 41.213
-.49 .204 24.201
-.49 1.0381 28.1305
-.49 2.2 32.06
-.49 3.60937 36.059
-.49 5.199 40.058
-.49 5.26214 40.2255
-.49 5.325 40.393
.196 .207 10.687
.196 .4439 12.3765
.196 .741 14.066
.196 1.2739 17.065
.196 1.881 20.064
.196 2.9 24.061
.196 4.249 28.058
.196 5.8592 32.0575
.196 7.529 36.057
.196 8.2218 37.69
.196 8.921 39.323
.49 .205 5.841
.49 .319 6.453
.49 .431 7.065
.49 .72482 8.5655
.49 .981 10.066
.49 1.3109 12.0655
.49 1.651 14.065
.49 1.847 15.0645
.49 2.031 16.064
.49 2.941 20.062
.49 4.1 24.06
.49 5.6094 28.058
.49 7.338 32.056
.49 8.8498 35.4695
.49 10.346 38.883
.98 .201 .831
.98 .32937 1.193
.98 .558 1.555
.98 .84406 2.309
.98 1.036 3.063
.98 1.2508 4.064
.98 1.441 5.065
.98 1.621 6.065
.98 1.811 7.065
.98 2.10725 8.565
.98 2.381 10.065
.98 2.7609 12.0645
.98 3.166 14.064
.98 3.38543 15.0635
.98 3.621 16.063
.98 4.70985 20.0605
.98 6.099 24.058
.98 7.78322 28.0565
.98 9.628 32.055
.98 11.0459 35.1215
.98 12.466 38.1880
1.569 .443 .031
1.569 .73779 .288
1.569 .98 .545

```

1.569 1.18262 .7975  
 1.569 1.365 1.05  
 1.569 1.667 1.553  
 1.569 1.918 2.056  
 1.569 2.30982 3.0585  
 1.569 2.63 4.061  
 1.569 2.911 5.062  
 1.569 3.171 6.063  
 1.569 3.41112 7.0635  
 1.569 3.641 8.064  
 1.569 4.081 10.064  
 1.569 4.521 12.064  
 1.569 4.79912 13.238  
 1.569 5.088 14.412  
 1.569 5.88294 17.2355  
 1.569 6.82 20.059  
 1.569 8.389 24.057  
 1.569 10.198 28.055  
 1.569 12.118 32.055  
 1.569 14.043 36.055  
 1.569 14.3628 36.724  
 1.569 14.682 37.393  
 1.961 .587 .031  
 1.961 .93975 .286  
 1.961 1.257 .541  
 1.961 1.53453 .7925  
 1.961 1.791 1.044  
 1.961 2.223 1.547  
 1.961 2.585 2.05  
 1.961 3.14751 3.054  
 1.961 3.569 4.058  
 1.961 3.91984 5.0595  
 1.961 4.22 6.061  
 1.961 4.491 7.062  
 1.961 4.741 8.063  
 1.961 5.221 10.063  
 1.961 5.701 12.063  
 1.961 6.0018 13.2375  
 1.961 6.31 14.412  
 1.961 7.17812 17.235  
 1.961 8.199 20.058  
 1.961 9.82922 24.0565  
 1.961 11.684 28.055  
 1.961 13.638 32.055  
 1.961 15.598 36.055  
 1.961 16.007 36.893  
 2.353 .593 .031  
 2.353 1.09723 .2845  
 2.353 1.528 .538  
 2.353 1.89568 .7895  
 2.353 2.217 1.041  
 2.353 2.776 11.544  
 2.353 3.233 2.047  
 2.353 3.936 3.051  
 2.353 4.466 4.055  
 2.353 4.8888 5.0575  
 2.353 5.254 6.06  
 2.353 5.56 7.061  
 2.353 5.841 8.062  
 2.353 6.361 10.062  
 2.353 6.891 12.062  
 2.353 7.20311 13.2365

2.353 7.545 14.411  
2.353 8.4412 17.2345  
2.353 9.489 20.058  
2.353 11.1882 24.0565  
2.353 13.085 28.055  
2.353 15.1576 32.224  
2.353 17.184 36.393  
3.137 .589 .031  
3.137 1.41 .283  
3.137 2.037 .535  
3.137 2.555 .786  
3.137 3.002 1.037  
3.137 3.7496 1.5395  
3.137 4.349 2.042  
3.137 5.3115 3.0455  
3.137 6.055 4.049  
3.137 6.672 5.052  
3.137 7.188 6.055  
3.137 7.629 7.057  
3.137 8.014 8.059  
3.137 8.6646 10.06  
3.137 9.24 12.061  
3.137 9.5983 13.236  
3.137 9.94 14.411  
3.137 10.8913 17.234  
3.137 11.969 20.057  
3.137 13.768 24.056  
3.137 15.608 28.055  
3.137 17.4539 31.764  
3.137 19.297 35.473  
3.431 .589 .031  
3.431 1.495 1.2825  
3.431 2.186 .534  
3.431 2.748 .785  
3.431 3.231 1.036  
3.431 4.0447 1.5385  
3.431 4.717 2.041  
3.431 5.7764 3.0445  
3.431 6.614 4.048  
3.431 7.3 5.0505  
3.431 7.877 6.053  
3.431 8.378 7.0555  
3.431 8.799 8.058  
3.431 9.4798 10.0595  
3.431 10.076 12.061  
3.431 10.431 13.2355  
3.431 10.795 14.41  
3.431 11.755 17.2335  
3.431 12.849 20.057  
3.431 14.563 24.056  
3.431 16.448 28.055  
3.431 18.196 31.594  
3.431 19.937 35.133  
4.118 .588 .031  
4.118 1.631 .282  
4.118 2.434 .533  
4.118 3.086 .784  
4.118 3.668 1.035  
4.118 4.6418 1.5365  
4.118 5.454 2.038  
4.118 6.782 3.041  
4.118 7.84 4.044

4.118 8.703 5.047  
4.118 9.415 6.05  
4.118 10.007 7.053  
4.118 10.508 8.056  
4.118 11.2796 10.058  
4.118 11.915 12.06  
4.118 12.2917 13.2345  
4.118 12.7 14.409  
4.118 13.7046 17.2335  
4.118 14.774 20.058  
4.118 16.4144 24.057  
4.118 18.198 28.056  
4.118 19.673 31.2245  
4.118 21.136 34.393  
4.51 .588 .031  
5.51 1.681 .282  
4.51 2.533 .533  
4.51 3.24766 .7835  
4.51 3.871 1.034  
4.51 4.938 1.5355  
4.51 5.842 2.037  
4.51 7.325 3.04  
4.51 8.499 4.043  
4.51 9.442 5.046  
4.51 10.214 6.049  
4.51 10.856 7.052  
4.51 11.388 8.055  
4.51 12.2086 10.057  
4.51 12.89 12.059  
4.51 13.2936 13.234  
4.51 13.719 14.409  
4.51 14.75 17.2335  
4.51 15.805 20.058  
4.51 17.369 24.057  
4.51 19.059 28.056  
4.51 20.3632 31.0245  
4.51 21.636 33.993  
4.902 .588 .031  
4.902 1.71 .282  
4.902 2.616 .533  
4.902 3.3854 .783  
4.902 4.056 1.034  
4.902 5.218 1.535  
4.902 6.21 2.036  
4.902 7.854 3.0385  
4.902 9.143 4.04  
4.902 10.1505 5.044  
4.902 10.974 6.048  
4.902 11.646 7.051  
4.902 12.218 8.054  
4.902 13.0988 10.0565  
4.902 13.829 12.059  
4.902 14.2515 13.233  
4.902 14.709 14.408  
4.902 15.7428 17.233  
4.902 16.769 20.059  
4.902 18.239 24.058  
4.902 19.814 28.057  
4.902 20.942 30.83  
4.902 22.05 33.603  
6.078 .587 .031  
6.078 1.771 .2815

6.078 2.791 .532  
6.078 3.69 .7825  
6.078 4.508 1.033  
6.078 5.953 1.5335  
6.078 7.217 2.034  
6.078 9.33 3.0365  
6.078 10.935 4.039  
6.078 12.129 5.042  
6.078 13.102 6.045  
6.078 13.899 7.048  
6.078 14.576 8.051  
6.078 15.6625 10.054  
6.078 16.529 12.057  
6.078 16.975 13.233  
6.078 17.405 14.409  
6.078 18.341 17.235  
6.078 19.21 20.06  
6.078 20.36 24.061  
6.078 21.525 28.061  
6.078 22.162 30.297  
6.078 22.789 32.533  
6.863 .587 .031  
6.863 1.8267 .2815  
6.863 2.92 .532  
6.863 3.988 .782  
6.863 4.792 1.033  
6.863 6.4045 1.5335  
6.863 7.825 2.034  
6.863 10.24 3.036  
6.863 12.048 4.038  
6.863 13.412 5.041  
6.863 14.49 6.044  
6.863 15.383 7.047  
6.863 16.115 8.05  
6.863 17.238 10.05  
6.863 18.086 12.058  
6.863 18.505 13.234  
6.863 18.9 14.41  
6.863 19.721 17.2365  
6.863 20.461 20.063  
6.863 21.3909 24.0635  
6.863 22.255 28.064  
6.863 22.644 29.9885  
6.863 23.017 31.913  
8.039 .587 .031  
8.039 1.8955 .2815  
8.039 3.079 .532  
8.039 4.16 .782  
8.039 5.157 1.032  
8.039 6.958 1.5325  
8.039 8.573 2.033  
8.039 11.33 3.0345  
8.039 13.45 4.036  
8.039 15.0745 5.039  
8.039 16.338 6.042  
8.039 17.352 7.046  
8.039 18.135 8.05  
8.039 19.248 10.055  
8.039 20.01 12.06  
8.039 20.357 13.2365  
8.039 20.671 14.413  
8.039 21.302 17.2395

|        |         |         |
|--------|---------|---------|
| 8.039  | 21.841  | 20.066  |
| 8.039  | 22.452  | 24.0675 |
| 8.039  | 22.922  | 28.069  |
| 8.039  | 23.074  | 29.591  |
| 8.039  | 23.212  | 31.113  |
| 8.823  | .587    | .031    |
| 8.823  | 1.965   | .2815   |
| 8.823  | 3.209   | .532    |
| 8.823  | 4.329   | .782    |
| 8.823  | 5.37    | 1.032   |
| 8.823  | 7.259   | 1.5325  |
| 8.823  | 8.943   | 2.033   |
| 8.823  | 11.835  | 3.034   |
| 8.823  | 14.079  | 4.035   |
| 8.823  | 15.833  | 5.038   |
| 8.823  | 17.197  | 6.041   |
| 8.823  | 18.271  | 7.045   |
| 8.823  | 19.105  | 8.049   |
| 8.823  | 20.258  | 10.055  |
| 8.823  | 21.0    | 12.061  |
| 8.823  | 21.301  | 13.238  |
| 8.823  | 21.566  | 14.415  |
| 8.823  | 22.0513 | 17.2415 |
| 8.823  | 22.442  | 20.068  |
| 8.823  | 22.827  | 24.069  |
| 8.823  | 23.177  | 28.07   |
| 8.823  | 23.253  | 29.3665 |
| 8.823  | 23.318  | 30.663  |
| 10.0   | .587    | .031    |
| 10.0   | 1.9484  | .2815   |
| 10.0   | 3.189   | .532    |
| 10.0   | 4.329   | .782    |
| 10.0   | 5.39    | 1.032   |
| 10.0   | 7.322   | 1.5325  |
| 10.0   | 9.062   | 2.033   |
| 10.0   | 12.085  | 3.034   |
| 10.0   | 14.498  | 4.035   |
| 10.0   | 16.407  | 5.0375  |
| 10.0   | 17.896  | 6.04    |
| 10.0   | 19.051  | 7.044   |
| 10.0   | 19.944  | 8.048   |
| 10.0   | 21.167  | 10.0545 |
| 10.0   | 21.915  | 12.061  |
| 10.0   | 22.197  | 13.239  |
| 10.0   | 22.422  | 14.417  |
| 10.0   | 22.761  | 17.2435 |
| 10.0   | 23.0    | 20.07   |
| 10.0   | 23.232  | 24.07   |
| 10.0   | 23.392  | 28.072  |
| 10.0   | 23.4195 | 29.0725 |
| 10.0   | 23.442  | 30.073  |
| 10.784 | .587    | .031    |
| 10.784 | 1.8596  | .2815   |
| 10.784 | 3.034   | .532    |
| 10.784 | 4.13    | .782    |
| 10.784 | 5.15    | 1.032   |
| 10.784 | 7.0292  | 1.5325  |
| 10.784 | 8.733   | 2.033   |
| 10.784 | 11.725  | 3.034   |
| 10.784 | 14.164  | 4.035   |
| 10.784 | 16.126  | 5.037   |
| 10.784 | 17.704  | 6.039   |

10.784 18.969 7.043  
 10.784 19.953 8.047  
 10.784 21.267 10.0535  
 10.784 22.075 12.06  
 10.784 22.381 13.2385  
 10.784 22.612 14.41  
 10.784 22.943 17.2435  
 10.784 23.142 20.07  
 10.784 23.362 24.071  
 10.784 23.482 28.072  
 10.784 23.494 28.9075  
 10.784 23.502 29.743  
 11.961 .588 .031  
 11.961 1.674 .2815  
 11.961 2.69 .532  
 11.961 3.647 .7825  
 11.961 4.548 1.033  
 11.961 6.222 1.533  
 11.961 7.764 2.033  
 11.961 10.515 3.034  
 11.961 12.917 4.035  
 11.961 15.02 5.0365  
 11.961 16.768 6.038  
 11.961 18.183 7.041  
 11.961 19.286 8.044  
 11.961 20.846 10.05  
 11.961 21.839 12.05  
 11.961 22.219 13.237  
 11.961 22.49 14.416  
 11.961 22.899 17.243  
 11.961 23.122 20.07  
 11.961 23.372 24.07  
 11.961 23.502 28.07  
 11.961 23.509 28.7325  
 11.961 23.512 29.393  
 12.941 .589 .031  
 12.941 1.435 .282  
 12.941 2.258 .533  
 12.941 3.043 .783  
 12.941 3.8 1.033  
 12.941 5.229 1.533  
 12.941 6.575 2.034  
 12.941 9.042 3.034  
 12.941 11.294 4.035  
 12.941 13.369 5.036  
 12.941 15.211 6.037  
 12.941 16.806 7.0395  
 12.941 18.118 8.042  
 12.941 20.04 10.0485  
 12.941 21.258 12.055  
 12.941 21.732 13.2345  
 12.941 22.081 14.414  
 12.941 22.574 17.24  
 12.941 22.882 20.069  
 12.941 23.222 24.07  
 12.941 23.462 28.071  
 12.941 23.488 28.657  
 12.941 23.512 29.243  
 14.02 .59 1.001  
 14.02 2.1554 1.517  
 14.02 3.66 2.033  
 14.02 6.326 3.034

14.02 8.663 4.035  
 14.02 10.76 5.0355  
 14.02 12.739 6.036  
 14.02 14.602 7.0375  
 14.02 16.244 8.039  
 14.02 18.732 10.045  
 14.02 20.266 12.05  
 14.02 20.87 13.23  
 14.02 21.316 14.41  
 14.02 21.981 17.239  
 14.02 22.432 20.067  
 14.02 22.962 24.068  
 14.02 23.382 28.069  
 14.02 23.437 28.641  
 14.02 23.492 29.213  
 15.098 .59 2.941  
 15.098 2.318 3.487  
 15.098 4.091 4.033  
 15.098 6.998 5.0335  
 15.098 9.546 6.034  
 15.098 11.783 7.035  
 15.098 13.83 8.036  
 15.098 17.1296 10.042  
 15.098 18.954 12.048  
 15.098 19.658 13.228  
 15.098 20.189 14.409  
 15.098 21.081 17.2365  
 15.098 21.731 20.064  
 15.098 22.551 24.065  
 15.098 23.261 28.066  
 15.098 23.393 28.6595  
 15.098 23.493 29.25  
 16.078 .588 5.067  
 16.078 4.447 6.05  
 16.078 7.762 7.033  
 16.078 12.144 8.535  
 16.078 15.153 10.037  
 16.078 17.441 12.045  
 16.078 18.677 14.053  
 16.078 19.138 15.056  
 16.078 19.524 16.059  
 16.078 20.769 20.06  
 16.078 22.004 24.06  
 16.078 22.7355 26.722  
 16.078 23.433 29.383  
 16.863 .587 6.801  
 16.863 3.559 7.417  
 16.863 6.27 8.032  
 16.863 12.98 10.036  
 16.863 15.967 12.04  
 16.863 16.83 13.04  
 16.863 17.42 14.05  
 16.863 17.955 15.05  
 16.863 18.419 16.056  
 16.863 19.969 28.058  
 16.863 21.34 24.06  
 16.863 22.236 26.821  
 16.863 23.074 29.583  
 17.647 .587 8.541  
 17.647 10.03 10.23  
 17.647 14.02 11.92  
 17.647 15.292 12.98

17.647 16.08 14.046  
17.647 16.74 15.05  
17.647 17.298 16.054  
17.647 18.958 20.0565  
17.647 20.39 24.059  
17.647 21.375 26.951  
17.647 22.309 29.843  
18.039 .586 9.406  
18.039 7.97 10.467  
18.039 12.014 11.528  
18.039 13.203 12.11  
18.039 14.027 12.708  
18.039 14.931 13.553  
18.039 15.649 14.399  
18.039 17.331 17.229  
18.039 18.399 20.058  
18.039 19.805 24.059  
18.039 21.15 28.06  
18.039 21.464 29.027  
18.039 21.774 29.993  
18.431 .585 10.276  
18.431 4.648 10.704  
18.431 8.007 11.132  
18.431 10.22 11.583  
18.431 11.64 12.034  
18.431 13.443 13.039  
18.431 14.63 14.044  
18.431 15.474 15.049  
18.431 16.087 16.053  
18.431 17.798 20.057  
18.431 19.165 24.06  
18.431 20.171 27.1065  
18.431 21.148 30.153  
18.824 .585 11.146  
18.824 5.835 11.589  
18.824 9.241 12.032  
18.824 11.024 12.453  
18.824 12.018 12.875  
18.824 13.29 13.635  
18.824 14.191 14.395  
18.824 15.54 16.2255  
18.824 16.414 18.056  
18.824 17.498 21.058  
18.824 18.47 24.06  
18.824 19.45 27.194  
18.824 20.419 30.328  
19.118 .584 11.797  
19.118 5.067 12.08  
19.118 8.005 12.36  
19.118 10.783 12.863  
19.118 12.018 13.365  
19.118 12.914 13.879  
19.118 13.565 14.392  
19.118 14.338 15.2215  
19.118 14.906 16.051  
19.118 15.903 18.0555  
19.118 16.64 20.06  
19.118 17.91 24.0605  
19.118 19.12 28.061  
19.118 19.48 29.282  
19.118 19.834 30.503  
19.608 .584 12.881

19.608 5.626 13.076  
19.608 8.004 13.271  
19.608 10.317 13.5625  
19.608 11.33 13.854  
19.608 11.944 14.122  
19.608 12.402 14.39  
19.608 13.355 15.269  
19.608 14.037 16.147  
19.608 15.747 20.104  
19.608 16.936 24.061  
19.608 17.853 27.39  
19.608 18.713 30.719  
19.804 .584 13.311  
19.804 4.002 13.406  
19.804 6.002 13.501  
19.804 8.3687 13.671  
19.804 10.006 13.841  
19.804 11.4077 14.149  
19.804 12.024 14.457  
19.804 13.25 15.63  
19.804 14.038 16.804  
19.804 15.479 20.433  
19.804 16.507 24.062  
19.804 17.412 27.443  
19.804 18.213 30.823  
20.0 .584 14.38  
20.0 4.4 14.561  
20.0 6.005 14.741  
20.0 8.273 15.0815  
20.0 10.01 15.422  
20.0 11.75 15.82  
20.0 13.339 16.217  
20.0 13.73 16.916  
20.0 14.038 17.615  
20.0 14.534 18.8375  
20.0 14.95 20.06  
20.0 16.1 24.062  
20.0 17.08 28.064  
20.0 17.4 29.4955  
20.0 17.703 30.927  
20.049 .584 15.542  
20.049 2.815 15.618  
20.049 4.003 15.694  
20.049 5.975 15.896  
20.049 7.402 16.097  
20.049 9.225 16.421  
20.049 10.764 16.746  
20.049 12.442 17.158  
20.049 13.821 17.57  
20.049 14.58 19.278  
20.049 15.126 20.987  
20.049 15.619 22.631  
20.049 16.049 24.276  
20.049 16.465 25.939  
20.049 16.852 27.602  
20.049 17.215 29.276  
20.049 17.563 30.95  
20.093 .584 16.586  
20.093 3.194 16.677  
20.093 4.375 16.768  
20.093 6.534 17.014  
20.093 8.138 17.26

20.093 10.138 17.654  
20.093 11.85 18.049  
20.093 13.102 18.407  
20.093 14.29 18.7658  
20.093 14.824 20.251  
20.093 15.268 21.736  
20.093 15.772 23.571  
20.093 16.237 25.405  
20.093 16.667 27.258  
20.093 17.062 29.11  
20.093 17.252 30.04  
20.093 17.437 30.971  
20.098 .584 16.7  
20.098 4.488 16.881  
20.098 6.005 17.061  
20.098 8.304 17.402  
20.098 10.009 17.742  
20.098 11.04 17.972  
20.098 12.01 18.202  
20.098 13.264 18.523  
20.098 14.322 18.849  
20.098 14.543 19.455  
20.098 14.75 20.06  
20.098 15.35 22.062  
20.098 15.89 24.063  
20.098 16.598 27.017  
20.098 17.23 29.97  
20.103 .584 16.819  
20.103 2.903 16.896  
20.103 4.055 16.97  
20.103 6.03 17.18  
20.103 7.504 17.388  
20.103 9.364 17.724  
20.103 10.912 18.06  
20.103 12.757 18.507  
20.103 14.267 18.954  
20.103 14.86 20.584  
20.103 15.37 22.214  
20.103 15.836 23.89  
20.103 16.26 25.566  
20.103 16.655 27.262  
20.103 17.008 28.959  
20.196 .584 19.032  
20.196 2.629 19.1  
20.196 3.779 19.168  
20.196 5.634 19.343  
20.196 6.956 19.519  
20.196 8.664 19.806  
20.196 10.1 20.09  
20.196 11.796 20.478  
20.196 13.204 20.862  
20.196 13.976 21.074  
20.196 14.746 21.285  
20.196 15.05 22.114  
20.196 14.34 22.943  
20.196 15.78 24.397  
20.196 16.17 25.851  
20.196 16.569 27.4  
20.196 16.947 28.951  
20.294 .584 21.361  
20.294 4.413 21.536  
20.294 6.005 21.711

20.294 8.225 22.049  
20.294 10.009 22.386  
20.294 13.064 23.1275  
20.294 15.414 23.869  
20.294 16.149 26.406  
20.294 16.884 28.943  
20.392 .588 23.67  
20.392 4.155 23.851  
20.392 5.975 24.031  
20.392 8.307 24.3715  
20.392 10.009 24.712  
20.392 12.188 25.2275  
20.392 14.012 25.743  
20.392 14.953 26.0365  
20.392 15.762 26.33  
20.392 16.284 27.637  
20.392 16.805 28.943  
20.49 .584 26.011  
20.49 3.117 26.086  
20.49 4.004 26.161  
20.49 7.627 26.599  
20.49 10.009 27.037  
20.49 12.125 27.535  
20.49 13.929 28.033  
20.49 15.086 28.4075  
20.49 16.131 28.782  
20.49 16.506 28.8625  
20.49 16.88 28.943

## APPENDIX B. ARVOL.M

```

%          arvol.m
%          by Russell wright
%          calculates station areas and center o' bouyancy
%          for T =16.9 ft

load hull1.dat
a=zeros(41,3);
zz=16.9
for i=1:41

    beg=23*(i-1)+1;
    endd=23*i;
    a(i,1)=hull1(beg);
    z=hull1(beg:endd,3);
    if z(1) > zz
        a(i,3)=0;a(i,2)=1;
    else
%   y=hull1(beg:endd,2);

        zp=hull1(beg:endd,3);
        z=zp(1):(16.9-zp(1))/22:16.9;
        yy=hull1(beg:endd,2);
        y=spline(zp,yy,z);
        h=z(2)-z(1);
        hhh(i)=h;
        four = 0.0;fr=0;
        two=0.0;to=0;

        for j= 1:11
            four=four+y(2*j);
            fr=fr+z(2*j)*y(2*j);
        end

        for k=1:10
            two= two+y(2*k+1);
            to=to+y(2*k+1)*z(2*k+1);
        end

        % calculate station area using simpson's rule:
        a(i,2)=2*(y(1)+4*four+2*two+y(23))*h/3;
        % calculate station centroid (z bar) using simpson's rule:
        a(i,3)=2*(y(1)*z(1)+4*fr+2*to+y(23)*z(23))*h/(3*a(i,2));

    end

end

aa=zeros(101,3);
hh=((a(37,1)-a(3,1))/100);
aa(:,1)=(a(3,1):hh:a(37,1))';
aa(:,2)=spline(a(3:37,1),a(3:37,2),aa(:,1));
aa(:,3)=spline(a(3:37,1),a(3:37,3),aa(:,1));

fore=0;too=0;fre=0;tw=0;fo=0;tu=0;
for i=1:50
    fore=fore+aa(2*i,2);
    fre=fre+aa(2*i,2)*aa(2*i,3);
    fo=fo+aa(2*i,2)*aa(2*i,1);
end

for i=1:49
    too=too+aa(2*i+1,2);

```

```

tw=tw+aa(2*i+1,2)*aa(2*i+1,3);
tu=tu+aa(2*i+1,2)*aa(2*i+1,1);
end

vol=(hh/3)*(aa(1,2)+4*fore+2*too+aa(101,2));
zbar=(aa(1,2)*aa(1,3)+4*fre+2*tw+aa(101,2)*aa(101,3))*hh/(3*vol)
xbar=(aa(1,2)*aa(1,1)+4*fo+2*tu+aa(101,2)*aa(101,1))*hh/(3*vol)

```

## APPENDIX C. DRAFTS.M

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%
%           drafts2.m
%           by russell wright
%           calculates wp points for a given draft
%

load hull1.dat
ind=0;
yincl=zeros(41,3);
zz=16.9;

zincl=zz
zinc2=zz
for i=1:41
i
    beg=23*(i-1)+1;
    endd=23*i;
    wpla(i,1)=hull1(beg,1);
    z=hull1(beg:endd,3);
    y=hull1(beg:endd,2);

    wpla(i,2)=interp1(z,y,zz,'spline');
    if z(1) > zz
        wpla(i,2)=0;
        yincl(i,2)=0;
        yincl(i,3)=0;
        wpla(i,1)=0;
    end

    yincl(i,1)=hull1(beg,1);
    yincl(i,2)=interp1(z,y,zincl,'spline');
    if yincl(i,2) < 0.0
        yincl(i,2)=0.0;
    end
    yincl(i,3)=zz;

    if wpla(i,2)==0
        yincl(i,2)=0;
        yincl(i,3)=0;
    end

end

yincl=yincl(3:37,:);

xx=yincl(:,1);
x=yincl(1,1):(yincl(35,1)-yincl(1,1))/34:yincl(35,1);
yy=yincl(:,2);
y=spline(xx,yy,x);

h=x(2)-x(1);
hhh(i)=h;
four = 0.0;fr=0;
two=0.0;to=0;

for j= 1:17
    four=four+y(2*j);
    fr=fr+x(2*j)*y(2*j);
end

```

```

end

for k=1:16
    two= two+y(2*k+1);
    to=to+y(2*k+1)*x(2*k+1);
end

% calculate wp 1/2 area using simpson's rule:
a2=2*(y(1)+4*four+2*two+y(35))*h/3;
% calculate wp centroid (x bar) using simpson's rule:
xbar=2*(y(1)*x(1)+4*fr+2*to+y(35)*x(35))*h/(3*a2)

% calculate long. moment o' inertia about xbar

fr=0;
to=0;

for j= 1:17
    fr=fr+(x(2*j)^2)*y(2*j);
end

for k=1:16
    to=to+y(2*k+1)*(x(2*k+1)^2);
end

io=2*(y(1)*x(1)*x(1)+4*fr+2*to+y(35)*(x(35)^2))*h/3;
isubl=io-a2*xbar^2
% calculate trans moment o' inertia

fr=0;
to=0;

for j= 1:17
    fr=fr+(y(2*j)^3);
end

for k=1:16
    to=to+(y(2*k+1)^3);
end

isubt=2*((y(1)^3)+4*fr+2*to+(y(35)^3))*h/9;

```

## LIST OF REFERENCES

1. Advanced Marine Enterprises "Advanced Stability Algorithm Phase I – Technology assessment", 1996
2. Dawson, P, "Computer Modeling of Ship Progressive Flooding as a Design Tool", Thesis, Naval Postgraduate School, Monterey, Ca. May 1995
3. Merritt, H. E., "Hydraulic Control Systems", John Wiley and Sons, Inc, 1967
4. Tupper, E., "Introduction to Naval Architecture", Butterworth-Heinemann, 1996



## INITIAL DISTRIBUTION LIST

|  | No. Copies |
|--|------------|
| 1. Defense Technical Information Center<br>8725 John J. Kingman Rd. STE 0944<br>Fort Belvoir, VA 22060-6218                                  | 2          |
| 2. Dudley Knox Library<br>Naval Postgraduate School<br>411 Dyer Road<br>Monterey, CA 93943-5101  | 2          |
| 3. Chairman, Code ME<br>Department of Mechanical Engineering<br>Naval Postgraduate School<br>Monterey, Ca 93942-5000                         | 2          |
| 4. Professor Fotis A. Papoulias, Code ME/ PA<br>Department of Mechanical Engineering<br>Naval Postgraduate School<br>Monterey, Ca 93942-5000 | 2          |
| 5. Curricular Officer, Code 34<br>Department of Mechanical Engineering<br>Naval Postgraduate School<br>Monterey, CA 93942-5000               | 1          |
| 6. LT Russell A. Wright, USN<br>Department Head Class 154<br>SWOSCOLCOM<br>446 Cushing Road<br>Newport, RI 02841-1209                        | 2          |